Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

Отчёт  
по лабораторной работе №2

«Работа с *JavaScript*»

|  |  |
| --- | --- |
| Проверил | Выполнил |
| Гончаревич А. Л. | ст. гр. №820602 |
|  | Д. С. Жуковский |

Минск 2021

**СОДЕРЖАНИЕ**

Введение 3

1 Цель работы 4

2 Теоретическая часть 5

3 Практическая часть 6

Заключение 7

Список использованных источников 8

# Введение

*JavaScript* был представлен в 1995 году как способ добавления программ на веб-страницы в браузере *Netscape Navigator*. С тех пор этот язык был принят другими основными браузерами. Это сделало возможными современные веб-приложения – приложения, с которыми можно взаимодействовать напрямую, не перезагружая страницу для каждого действия. *JavaScript* также используется на более традиционных веб-сайтах для обеспечения различных форм интерактивности [1].

Важно отметить, что *JavaScript* почти не имеет ничего общего с языком программирования *Java*. Изначально первый имел название *Mocha*. Брендан Эйх создал его в 1992 году. В это же время компания *Sun* *Microsystems* заканчивала разработку *Java*. Эйх договорился с ними – и *Mocha* стал скриптовым компаньоном языка *Java*. Познее *Mocha* был переименован в *LiveScript*, что также не принесло языку славы. Тогда, воспользовавшись популярностью своего напарника, разработчики дали ему название *JavaScript*.

*JavaScript* используют в основном в веб-разработке. Вместе с *HTML* и *CSS*, *JavaScript* создаёт идеальный набор для фронтенд-разработчика.

В веб-разработке *JavaScript* помогает обрабатывать любое действие пользователя – от заполнения формы до прокрутки страницы. В браузере срабатывает событие, которое запускает работу *JavaScript*-кода, только после этого на странице происходит изменение. На каждом сайте используется несколько скриптов, которые собирают данные посетителей, анализируют метрики и позволяют зарегистрировать аккаунт.

*JavaScript* используют также для следующих целей:

– разработка нативных приложений;

– бэкенд-разработка;

– создание ПО для техники;

– разработка десктопных приложений;

– создание расширений для браузера.

*JavaScript* – постоянно развивающийся язык с продуманной инфраструктурой и большой экосистемой. У этого языка нет конкурентов в его основной сфере применения, он справляется со всеми поставленными задачами. Поэтому сейчас нет необходимости в создании альтернативного варианта для него – он достаточно быстр, удобен и универсален.

Также для этого языка существуют фреймворки, в которых есть предварительно написанный код для использования в функциях и задачах программирования. Наиболее популярны фреймворки *React*, *Angular*, *Vue*.

# Цель работы

Познакомиться с жаваскрипт.

Сделать задания с использованием жс.

# Теоретическая часть

## Использование *JavaScript* на веб-страницах

*JavaScript*-код может встраиваться в *HTML*-документы следующими способами:

– встроенные сценарии между парой тегов <*script*> и </*script*>;

– из внешнего файла, заданного атрибутом *src* элемента *script*;

– в обработчик события, заданный в качестве значения *HTML*-атрибута;

– как тело *URL*-адреса.

*HTML*-атрибуты обработчиков событий и адреса *URL* с псевдопротоколом *javascript* редко используются в современной практике программирования на языке *JavaScript* (они были более распространены на раннем этапе развития Интернета).

Встроенные сценарии также стали реже использоваться по сравнению с прошлым. Согласно философии программирования, известной как ненавязчивый *JavaScript* (*unobtrusive* *JavaScript*), содержимое (разметка *HTML*) и поведение (программный код на языке *JavaScript*) должны быть максимально отделены друг от друга. Следуя этой философии программирования, сценарии на языке *JavaScript* лучше встраивать в *HTML*-документы с помощью элементов *script*, имеющих атрибут *src*. Элемент *script* с атрибутом *src* ведёт себя точно так, как если бы содержимое указанного файла *JavaScript*-кода находилось непосредственно между тегами <*script*> и </*script*>.

При использовании атрибута *src* любое содержимое между открывающим и закрывающим тегами <*script*> игнорируется. При желании в качестве содержимого можно вставлять описание включаемого программного кода или информацию об авторском праве. Однако следует заметить, что инструменты проверки соответствия разметки требованиям стандарта *HTML*5 будут выдавать предупреждения, если между тегами <*script*> и </*script*> с атрибутом *src* будет находиться какой-либо текст, не являющийся пробельными символами или комментариями на языке *JavaScript*.

Использование тега *script* с атрибутом *src* даёт ряд преимуществ:

1. *HTML*-файлы становятся проще, так как из них можно убрать большие блоки *JavaScript*-кода. Это помогает отделить содержимое от поведения.
2. *JavaScript*-функцию или другой код, используемый несколькими *HTML*-документами, можно хранить в одном файле и считывать при необходимости. Это уменьшает объём занимаемой дисковой памяти и намного облегчает поддержку программного кода, так как отпадает необходимость править каждый файл при изменении кода.
3. Если сценарий на языке *JavaScript* используется сразу несколькими страницами, он будет загружаться браузером только один раз, при первом его использовании – последующие страницы будут извлекать его из кэша браузера.
4. Атрибут *src* принимает в качестве значения произвольный *URL*-адрес, поэтому *JavaScript*-программа или веб-страница с одного веб-сервера может воспользоваться кодом (например, из библиотеки подпрограмм), предоставляемым другими веб-серверами. Многие рекламодатели в Интернете используют этот факт.

Обработчик кнопки на главной странице

Кастомная функция и передача параметров и онклик

Объект виндоу

Confirm

Объект документ

Установка стилей (шрифта из кода)

Алерт и промпт (встроенные методы)

Конкатенация строк

Калькулятор

Ифы форы вар

Динамическая страница

Массивы создание элементов из жс

Встраивание элементов в родительские

свитч

# Практическая часть

## Описание используемых программных средств

### Visual Studio Code…

Кроссплатформенный, быстрый, есть лайв сервер.

## Описание назначений фреймов

### Начальная страница служит для первоначального ознакомления пользователя со страницей, содержащей фреймы для демонстрации необходимого функционала. Здесь можно найти демонстрацию методов confirm, prompt и alert, а также калькулятор и словарь. На страницу можно попасть со страницы «Учёба» по гиперссылке «Прочее» и вернуться обратно с помощью кнопки «Вернуться». Данная страница приведена на рисунке 2.

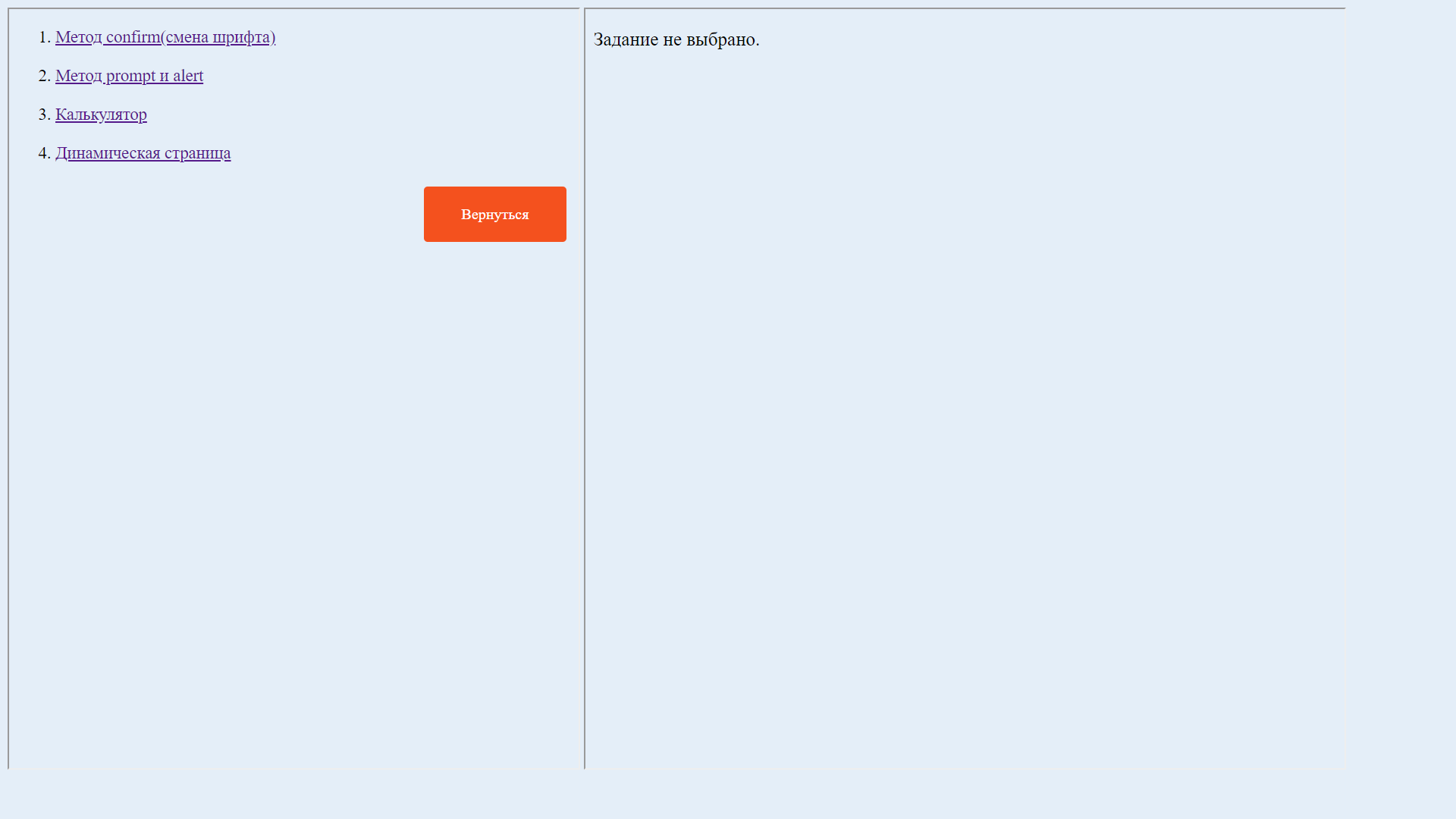


Рисунок 2 – Начальная страница

### Фрейм «Метод confirm(смена шрифта)» служит для демонстрации работы метода confirm и реализует смену шрифта данного фрейма со стандартного (Times New Roman в браузере Chrome) на шрифт Arial. Данная страница приведена на рисунке 3.

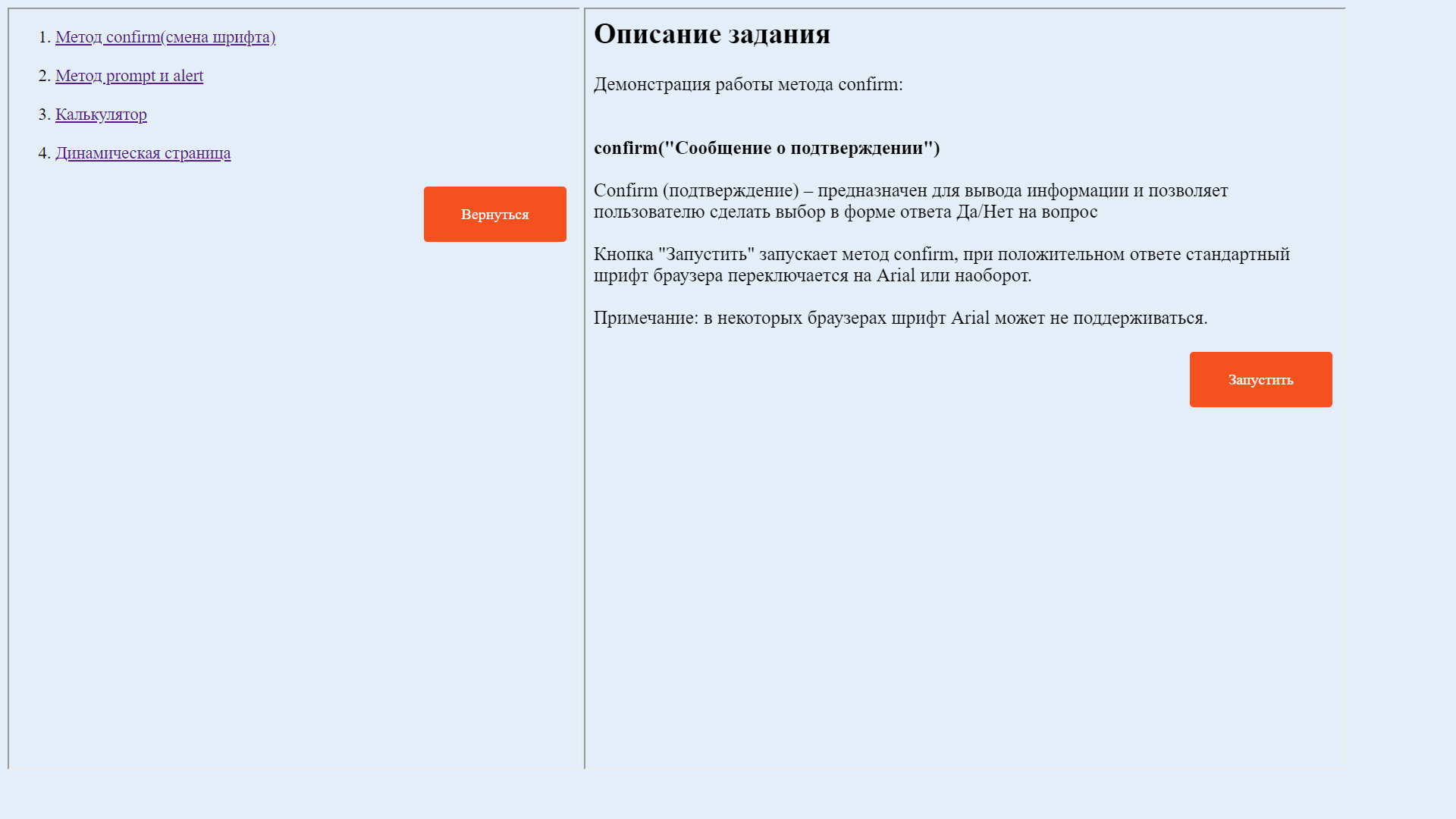


Рисунок 3 – Фрейм «Метод *confirm*(смена шрифта)»

### Фрейм «Метод *prompt* и *alert*» служит для демонстрации работы методов *prompt* и *alert* и реализует их работу посредством взаимодействия с пользователем. Данная страница приведена на рисунке 4.

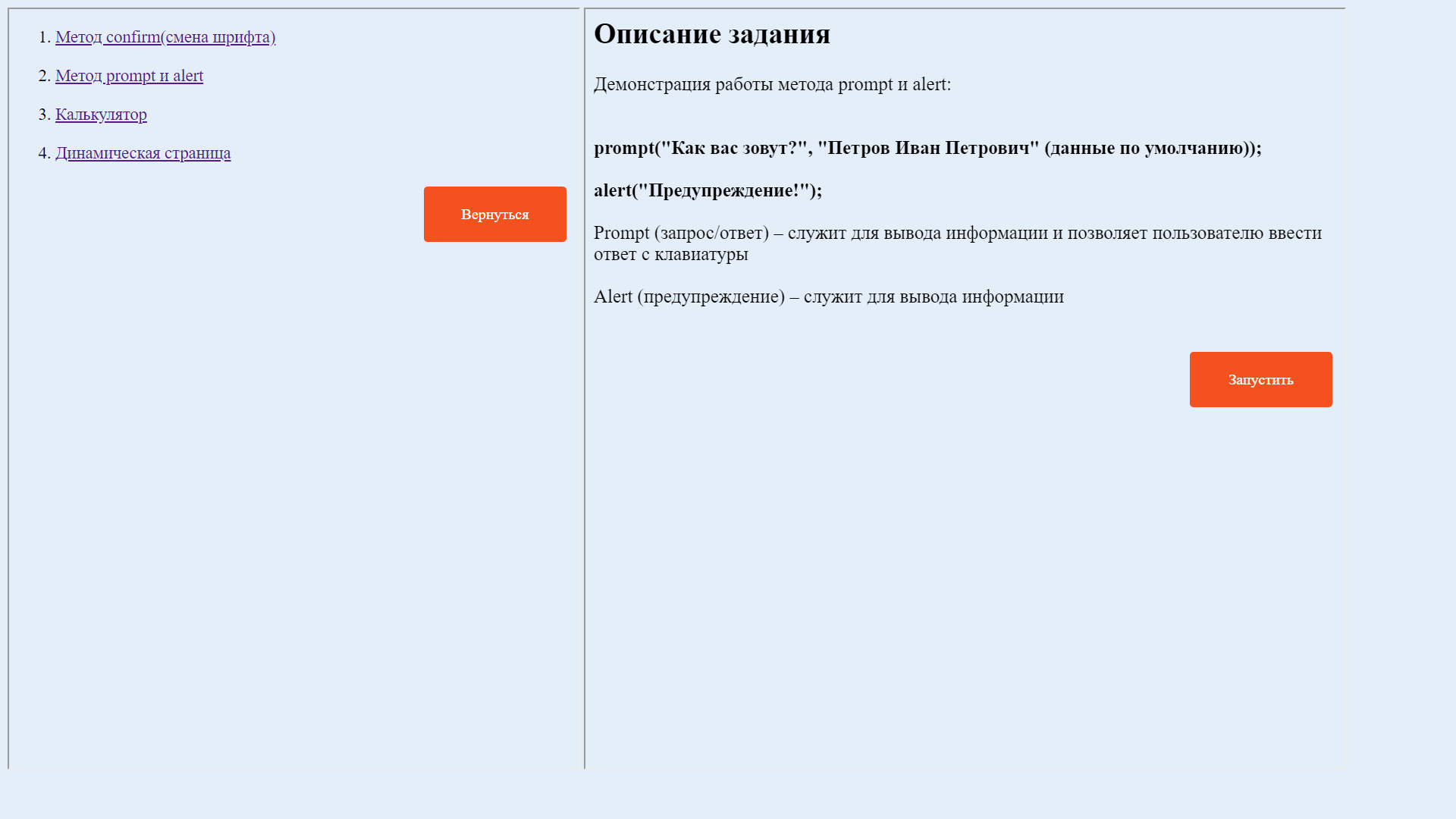


Рисунок 4 – Фрейм «Метод *prompt* и *alert*»

### Фрейм «Калькулятор» предоставляет пользователю достаточно большой функционал для подсчёта различных значений. Пользователь может складывать, вычитать, делить и умножить как обычные числа, так и десятичные. Также есть возможность вычислить инкремент или декремент текущего числа. Есть ещё функции сброса, удаления последнего символа, смены знака и получения значения одного процента от текущего числа (т.е. его деление на 100). Выше числового поля есть текстовое поле, по возможности показывающее последнюю проделанную операцию и последнее полученное значение. Ввод производится только с помощью кнопок. Данная страница приведена на рисунке 5.

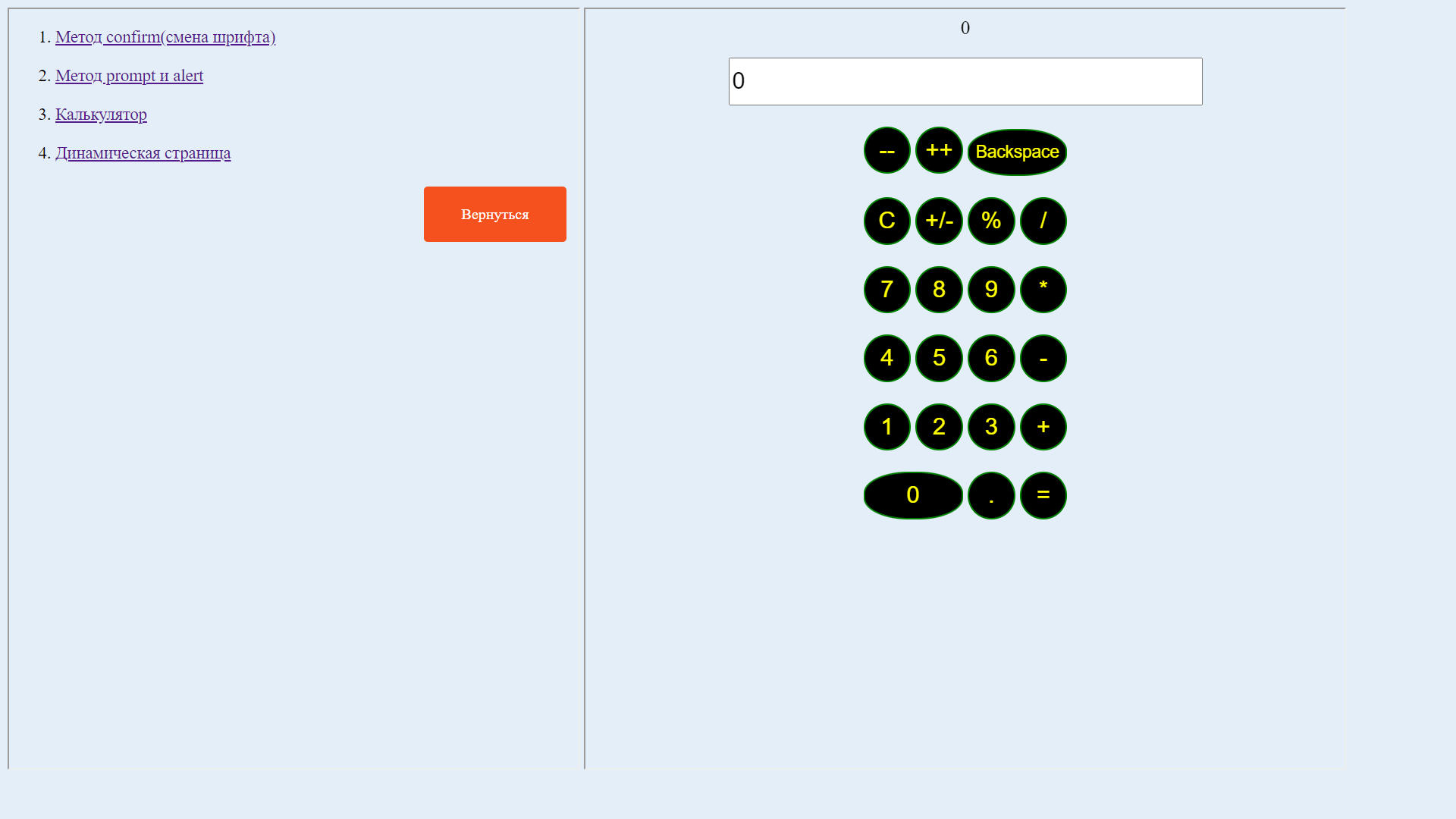


Рисунок 5 – Фрейм «Калькулятор»

### Фрейм «Динамическая страница» представляет собой словарь с набором из десяти слов, которые можно перевести на три языка: белорусский, английский и французский. Перевод производится посредством перерисовки таблицы в зависимости от выбранного переключателя. Данная страница приведена на рисунке 6.

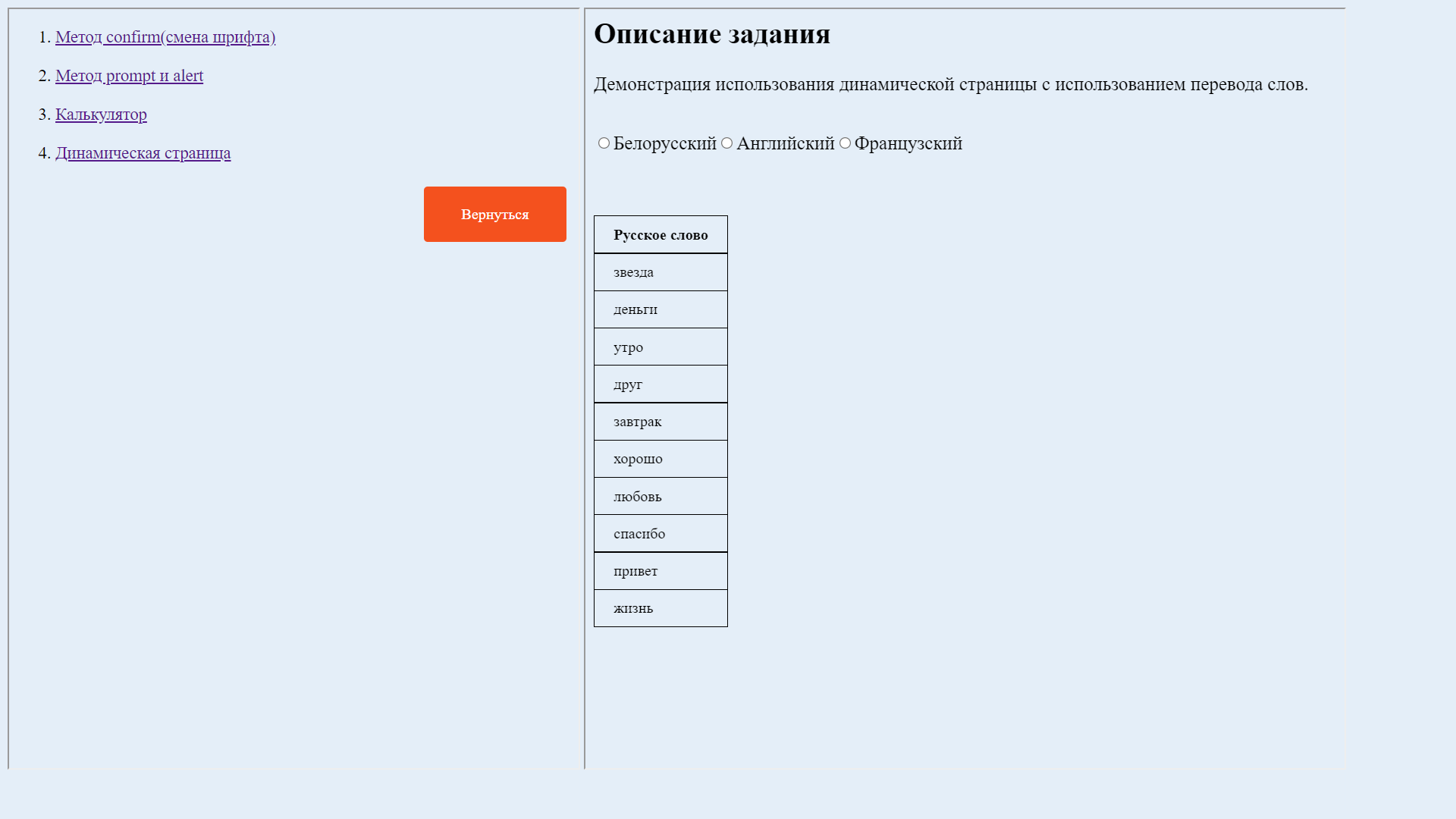


Рисунок 6 – Фрейм «Динамическая страница»

## Руководство пользователю

Пользование функционалом начинается со стартовой страницы, на которой располагается четыре ссылки на страницы, загружаемые в соседний фрейм. Кнопка «Вернуться» вернёт пользователя на страницу «Учёба».

Чтобы протестировать функции первых двух ссылок – «Метод *confirm*(смена шрифта)» и «Метод *prompt* и *alert*» следует после непосредственного перехода по ним нажать кнопку «Запустить». В случае с методами *prompt* и *alert* пользователя дополнительно надо будет ввести нужный ему текст и нажать кнопку «ОК» или «ОТМЕНА», если пользователь выбрал опцию «ОК», далее также следует нажать кнопку «ОК» для подтверждения всплывшей информации.

Для пользования калькулятором следует нажимать кнопки, предоставленные пользователю.

Для смены языка функции «Динамическая страница» следует выбрать один из переключателей перевода слов на нужный пользователю язык.

# Заключение

# Список использованных источников

[1] Eloquent JavaScript / Marijn Haverbeke, 2018.

[3] *Notepad++* [Электронный ресурс]. – Режим доступа : *https://notepad-plus-plus.org*/*.*